

Resume Ranking System

Ms Rajeshwari¹, A Aashritha², M Bhavani³, R Bhavani⁴

¹Assistant Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.

^{2,3,4}B.Tech Students; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; bhavanirapolu14@gmail.com

Accepted 14-06-2026

Author(s) Retains the Copyrights of This Article

Abstract

The Fair Hire AI System is designed to support fair, transparent, and efficient recruitment by leveraging Artificial Intelligence (AI) and Machine Learning (ML) techniques. Traditional hiring processes are often influenced by human bias and time-consuming manual screening, which can impact the fairness and accuracy of candidate selection. The proposed system addresses these challenges by evaluating applicants solely on the basis of their skills, qualifications, and professional experience while eliminating sensitive personal information such as name, gender, and location to ensure an unbiased assessment process. Candidates can upload their resumes, which are analyzed using Natural Language Processing (NLP) techniques to extract relevant information and compare it with job requirements. Based on this analysis, the system generates candidate rankings and presents recruiters with a structured and user-friendly view of potential applicants through an intuitive interface. Additionally, the system ensures secure handling of candidate data and facilitates efficient recruitment management. By reducing manual effort, improving selection accuracy, and promoting fairness, transparency, and inclusivity, the Fair Hire AI System contributes to a more effective and equitable hiring process.

Keywords— Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), Fair Hiring, Recruitment System, Resume Screening, Candidate Ranking, Bias Reduction, Talent Acquisition, Data Security.

Introduction

The rapid growth of online recruitment platforms has significantly increased the number of job applications received by organizations. As a result, recruiters often face challenges in reviewing and evaluating large volumes of resumes within limited timeframes. Conventional recruitment methods rely heavily on manual resume screening, which is labor-intensive, time-consuming, and susceptible to subjective judgment. To address these limitations, intelligent recruitment systems have emerged as an effective solution for automating candidate evaluation and selection.

The Resume Ranking System is an Artificial Intelligence (AI)-driven application developed to streamline the hiring process by automatically analyzing and ranking candidate resumes according to job-specific requirements. The system examines candidate profiles, including skills, educational qualifications, professional experience, and technical competencies, and compares them with the requirements specified in a job description. Based on this analysis, a ranking score is generated for each applicant, enabling recruiters to identify the most suitable candidates quickly and efficiently. The primary objective of the proposed system is to reduce

recruitment effort, enhance selection accuracy, and accelerate the overall hiring process through automation.

Existing Recruitment System

Most organizations continue to depend on manual resume screening performed by Human Resource (HR) professionals. In this approach, recruiters individually review each submitted resume and determine candidate suitability based on their qualifications and experience. Although widely used, this method becomes increasingly inefficient when the number of applications is large. Manual evaluation often results in inconsistent assessments, overlooked talent, and delayed recruitment decisions. Furthermore, personal preferences and unconscious biases may influence the selection process, affecting fairness and objectivity. Consequently, traditional recruitment systems struggle to maintain consistency, speed, and accuracy in candidate evaluation.

Proposed System

The proposed Resume Ranking System introduces an AI-enabled recruitment framework that automates candidate screening and ranking. The system employs Machine Learning (ML) and Natural Language

Processing (NLP) techniques to analyze resumes and identify qualified applicants based on predefined job requirements. By extracting and evaluating relevant information such as technical skills, educational background, certifications, and work experience, the system performs objective candidate assessment with minimal human intervention.

Requirement Analysis

Requirement analysis was conducted to identify the functional and non-functional needs of the Resume Ranking System. The system is designed to provide recruiters with an efficient platform for candidate evaluation while ensuring scalability, usability, and maintainability.

Functional Requirements

The recruitment module allows recruiters to create accounts, authenticate into the system, upload resumes in PDF or DOC formats, specify job requirements, define ranking thresholds, view ranked candidates, export shortlisted applicants, and securely log out. The Resume Ranking System performs user authentication, resume processing, text extraction, candidate profile analysis, personal information removal, job-resume matching, similarity score computation, and automated candidate ranking.

Non-Functional Requirements

The system is designed to efficiently process large numbers of resumes while maintaining high performance and reliability. Usability considerations ensure that recruiters can interact with the application through a simple and intuitive interface. Maintainability is achieved through modular system architecture, allowing future enhancements and updates. Portability enables deployment across different computing environments, while performance requirements ensure fast and accurate candidate ranking.

Computational Resource Requirements

The proposed system requires a computer equipped with an Intel Core i5 processor, 8 GB RAM, and 500 GB storage capacity. The software environment includes Windows 10, Python 3.12, HTML, CSS, JavaScript, MySQL, and supporting libraries such as NumPy and Pandas. Development and testing activities are performed using Visual Studio Code.

System Development Methodology

The development of the Resume Ranking System follows the Waterfall Software Development Life Cycle (SDLC) model. This methodology provides a structured and sequential approach to software development, ensuring systematic progress through each phase.

The requirement analysis phase focuses on understanding recruitment challenges and defining system objectives. During the design phase, the architecture of the resume parser, ranking engine, database components, and user interface is established. The implementation phase involves coding and integrating individual modules, including resume upload, text extraction, candidate matching, and ranking functionalities.

Following implementation, comprehensive testing is conducted to verify system functionality, ranking accuracy, and overall performance. Finally, the deployment and maintenance phase involves system installation, monitoring, bug resolution, and continuous enhancement to ensure reliable operation and long-term usability.

SYSTEM DESIGN

System design defines the structure and operational framework of the Resume Ranking System. The architecture ensures smooth interaction between users, processing modules, and databases.

Software Architecture

The software architecture consists of multiple layers, including the user interface, application processing layer, resume analysis module, ranking engine, and database layer. These components work together to automate candidate evaluation and ranking.

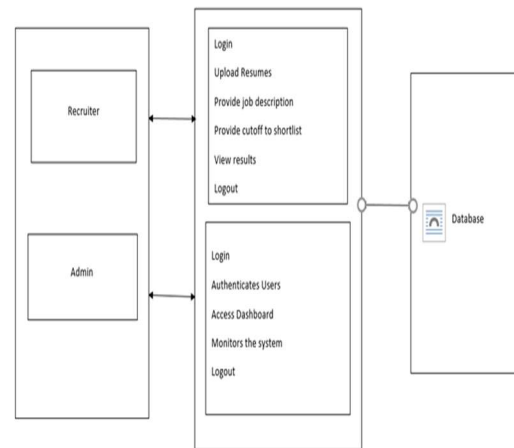


Fig 1; Software Architecture

Technical Architecture

The technical architecture integrates frontend technologies, backend processing modules, machine learning algorithms, NLP techniques, and database management systems. This architecture enables efficient resume parsing, candidate assessment, and ranking generation while ensuring scalability and security.

large volumes of resumes while maintaining high performance and reliability.



Fig 2; Technical Architecture

Database Design

The database is one of the most important components of the Resume Ranking System, as it is responsible for storing, organizing, and managing all recruitment-related information. A relational database structure is used to maintain data integrity, security, and efficient access to records. The database stores recruiter details, candidate information, uploaded resumes, job descriptions, and ranking results generated by the system. Proper database design ensures smooth communication between the application modules and supports efficient processing of recruitment data. The database also enables quick retrieval of information required for resume analysis, candidate evaluation, and ranking generation.

The ranking database maintains the results produced by the candidate evaluation process. After analyzing resumes and comparing them with job requirements, the system calculates similarity scores and stores them along with candidate details and ranking positions. This information allows recruiters to review candidate performance and identify the most suitable applicants. The ranking results database further stores shortlisted candidates and their final scores, providing a structured view of recruitment outcomes. These records assist recruiters in making objective and data-driven hiring decisions.

The recruiter database contains information related to registered recruiters, including login credentials, profile details, and system access permissions. Secure authentication mechanisms are implemented to protect recruiter accounts and ensure that only authorized users can access recruitment data. The resume database stores uploaded resumes and extracted textual information obtained through Natural Language Processing techniques. Candidate details such as skills, educational qualifications, certifications, and work experience are maintained within the database for further analysis. Efficient database management enables the system to process

```
mysql> describe ranking;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
recruiter_id	int	NO	MUL	NULL	
job_description	text	NO		NULL	
cutoff_score	float	YES		NULL	
created_at	datetime	YES		NULL	

Fig. 3; Ranking

```
mysql> describe ranking_results;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
ranking_id	int	YES	MUL	NULL	
resume_id	int	YES	MUL	NULL	
score	decimal(5,2)	YES		NULL	
candidate_rank	int	YES		NULL	
status	enum('Shortlisted', 'Discarded')	YES		Discarded	

Fig. 4; Ranking Results

```
mysql> describe resumes;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
recruiter_id	int	YES	MUL	NULL	
filename	varchar(255)	NO		NULL	
original_text	longtext	YES		NULL	
cleaned_text	longtext	YES		NULL	
extracted_skills	text	YES		NULL	
file_path	varchar(500)	YES		NULL	
upload_date	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

Fig. 5; Resumes

Implementation

Implementation is the phase in which the designed Resume Ranking System is transformed into a functional application. The system is developed using modern web technologies and intelligent data-processing techniques to automate resume screening and candidate evaluation. The implementation integrates a user-friendly web interface with backend processing modules and Artificial Intelligence-based ranking mechanisms. By combining Machine Learning and Natural Language Processing techniques, the system is capable of extracting relevant information from resumes, comparing candidate profiles with job requirements, and generating ranking scores automatically. This approach significantly reduces manual effort and improves the efficiency of the recruitment process.

The system is primarily developed using Python, which serves as the core programming language for implementing application logic and AI functionalities. The Flask framework is used to build the backend architecture of the application. Flask provides a lightweight and flexible environment for handling user requests, processing uploaded resumes, managing candidate information, and communicating with the database. Its integration capabilities make it suitable for developing intelligent recruitment applications that require seamless interaction between multiple components.

The frontend of the application is developed using HTML, CSS, and JavaScript. HTML provides the structural framework for web pages, including login forms, resume upload interfaces, recruiter dashboards, and ranking result displays. CSS is used to enhance the visual appearance of the application by creating a responsive and user-friendly interface. JavaScript adds dynamic functionality to the system by enabling form validation, interactive features, and real-time updates that improve the overall user experience.

MySQL is utilized as the database management system for storing and managing recruitment-related information. The database maintains records of recruiters, candidate profiles, resumes, job descriptions, and ranking results. Secure data storage and efficient retrieval mechanisms ensure smooth system operation and support large-scale resume processing. The integration of MySQL with the backend application facilitates reliable data management and contributes to the overall performance and scalability of the Resume Ranking System.

TESTING

Software testing is an essential phase in the software development life cycle that evaluates the functionality, reliability, performance, and security of an application. The primary objective of testing is to identify defects and verify that the developed system satisfies the specified requirements. In the proposed Resume Ranking System, testing is performed to ensure the accurate processing of resumes, effective candidate evaluation, and proper execution of all system functionalities. Various modules, including user authentication, resume upload, text extraction, candidate ranking, and result generation, are thoroughly verified during the testing process.

Since the system is designed to support automated recruitment and candidate assessment, ensuring fairness, accuracy, and reliability is of significant importance. Testing is conducted to validate the correctness of resume analysis, similarity scoring, and ranking mechanisms while maintaining the security and confidentiality of candidate information. A well-tested system improves recruiter confidence, enhances user satisfaction, and contributes to better recruitment outcomes. Furthermore, testing helps identify performance bottlenecks and ensures that the system can handle multiple resumes efficiently without affecting processing accuracy.

The testing process contributes to improved candidate matching accuracy, reliable system operation, protection of sensitive recruitment data, and enhanced overall application quality. Through systematic testing procedures, the system achieves higher performance, stability, and consistency in candidate evaluation.

Dimensions of Testing

Several dimensions of testing are considered to ensure comprehensive validation of the Resume Ranking System. Testing is performed across different layers of the application, including the user interface, backend processing modules, and database management components. This approach helps verify that each layer functions correctly and interacts efficiently with other parts of the system.

The testing strategy also considers multiple testing levels, including unit testing, integration testing, and system testing. These levels enable verification of individual components as well as the complete application. Different testing categories, such as functional testing, performance testing, and security testing, are employed to evaluate system behavior under various conditions. In addition, both manual testing and automated validation techniques are utilized to identify defects and confirm the correctness of system operations.

Stages of Testing

Unit Testing

Unit testing is performed to verify the functionality of individual modules within the Resume Ranking System. Each component is tested independently to ensure that it performs its intended task correctly. Modules such as user authentication, resume parsing, keyword extraction, similarity calculation, and candidate ranking are validated separately. This testing phase helps identify coding errors and logical issues at an early stage of development, thereby reducing the complexity of future debugging activities.

Integration Testing

Integration testing is conducted after individual modules have been successfully tested. In this stage, multiple components are combined and evaluated as an integrated system. The interaction between modules such as resume upload, Natural Language Processing, candidate evaluation, database storage, and result generation is verified. The primary objective of integration testing is to ensure smooth data flow and proper communication among different system components. This testing phase helps identify interface-related issues and ensures that all modules work together effectively.

Types of Testing

Black Box Testing

Black Box Testing is performed to evaluate the functionality of the Resume Ranking System without examining the internal program structure or source code. In this testing approach, the system is treated as a closed unit, and validation is performed solely on the basis of inputs and outputs. Different resumes, job descriptions, and user requests are provided as inputs,

and the generated results are analyzed to verify correctness and consistency.

The objective of black box testing is to ensure that the system fulfills all functional requirements and produces accurate outputs for valid inputs while appropriately handling invalid or unexpected data. This testing method helps identify missing functionalities, incorrect processing results, and user interface issues. Since the testing process focuses on the user perspective, it provides valuable insights into the usability and effectiveness of the application.

White Box Testing

White Box Testing is carried out to examine the internal structure, program logic, and execution flow of the Resume Ranking System. This testing approach focuses on verifying source code implementation, control structures, conditional statements, loops, and algorithm execution. The internal operations of modules such as resume parsing, skill extraction, similarity computation, and candidate ranking are carefully evaluated to ensure correct functionality.

White box testing helps identify logical errors, inefficient code segments, and potential vulnerabilities within the system. It also verifies that different modules interact correctly and that all execution paths are properly tested. By analyzing the internal behavior of the application, this testing technique improves software quality, enhances performance, and increases overall system reliability. As a result, the Resume Ranking System operates efficiently and delivers accurate candidate evaluation results.

Screenshots

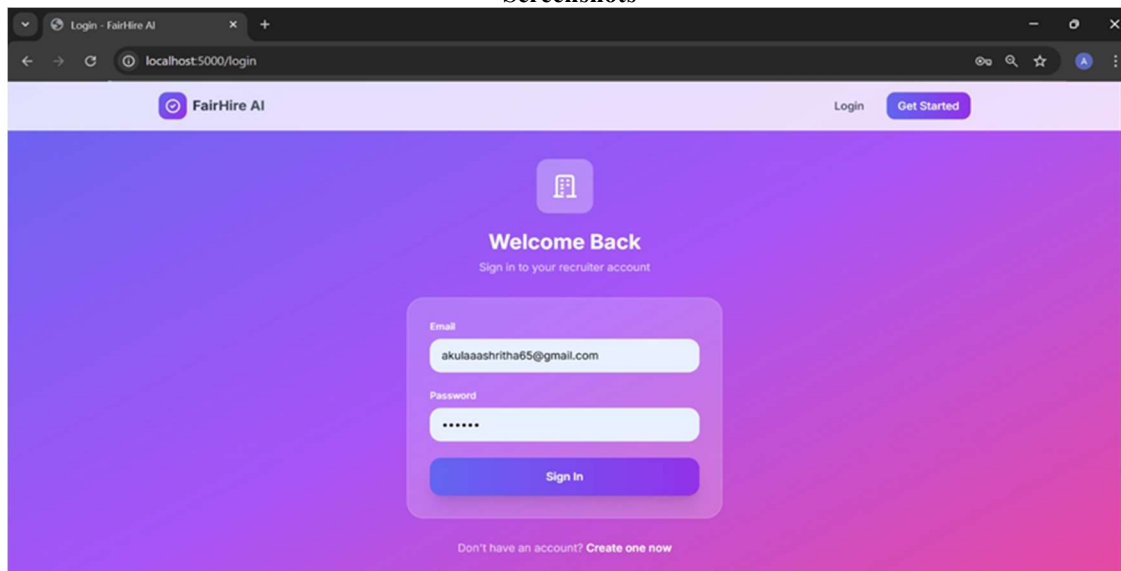


Fig.6; Login Page

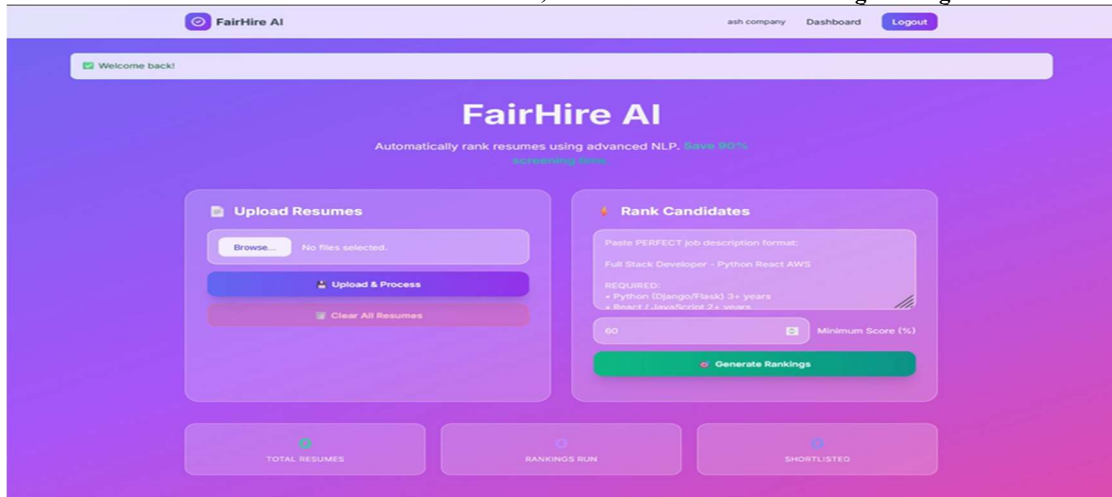


Fig.7; Dashboard

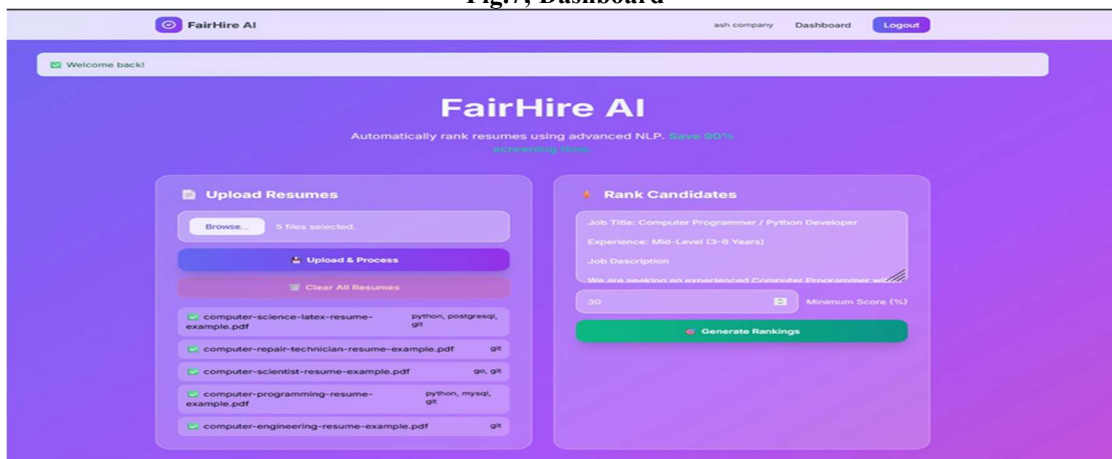


Fig. 8; Resumes uploaded

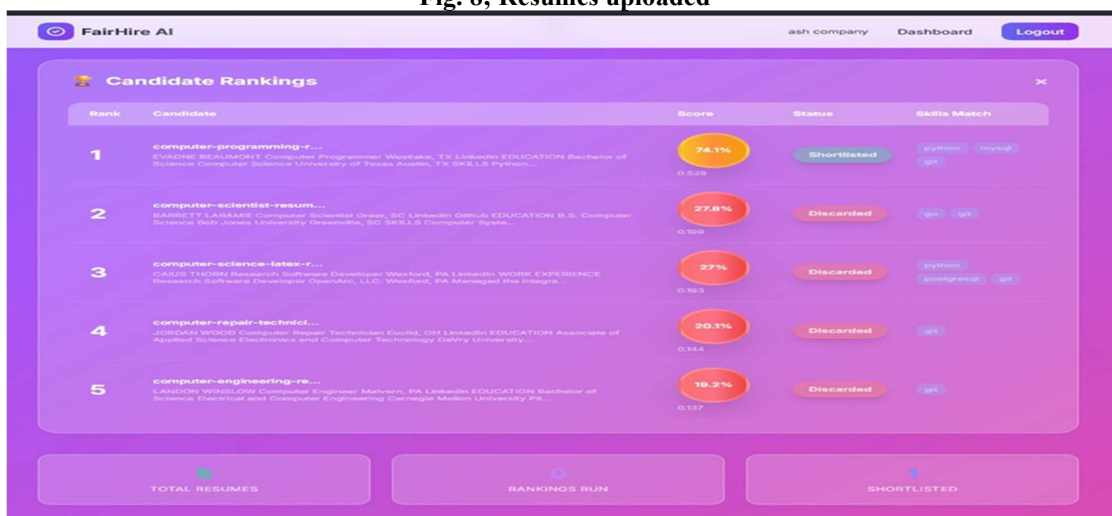


Fig. 9; Ranked Resume

Conclusion

The Resume Ranking System provides an intelligent and efficient approach to modern recruitment by

automating the process of resume screening and candidate evaluation. The system utilizes Artificial Intelligence, Machine Learning, and Natural

Language Processing techniques to analyze candidate resumes and compare them with job requirements. By extracting relevant information such as skills, educational qualifications, certifications, and professional experience, the system generates ranking scores that assist recruiters in identifying the most suitable candidates.

The implementation of automated resume analysis significantly reduces the time and effort required for manual screening while improving the accuracy and consistency of candidate selection. Furthermore, the system minimizes the possibility of human errors and subjective bias by relying on objective evaluation criteria. The integration of a user-friendly interface, efficient database management, and intelligent ranking mechanisms enhances the overall recruitment process and supports data-driven hiring decisions. Therefore, the proposed Resume Ranking System serves as a reliable, scalable, and effective solution for organizations seeking to improve recruitment efficiency and talent acquisition outcomes.

Future Scope

The proposed system offers several opportunities for future enhancement and expansion. One potential improvement is the integration of professional recruitment platforms and job portals such as LinkedIn, Indeed, and Naukri.com. Such integration would enable automatic retrieval of candidate profiles and resume data, eliminating the need for manual resume uploads and improving the efficiency of the recruitment process.

Future versions of the system may also incorporate advanced Artificial Intelligence models capable of performing deeper candidate analysis, including skill gap assessment, career progression evaluation, and job-fit prediction. The inclusion of real-time job posting and applicant tracking functionalities would further enhance recruiter productivity. Additionally, cloud-based deployment and big-data processing capabilities can be implemented to support large-scale recruitment operations involving thousands of applications.

Another promising enhancement involves the use of explainable AI techniques to provide transparent candidate ranking justifications. This feature would help recruiters understand the reasoning behind ranking decisions and increase trust in automated hiring systems. Furthermore, the integration of interview scheduling, candidate communication modules, and predictive workforce analytics could transform the system into a comprehensive end-to-end recruitment management platform capable of supporting future organizational hiring requirements.

References

- [1] C. Sai Surya, G. Kiriti, K. Satya Saketh, P. Sri Charan, and A. Anjali, "Optimizing Job Matching Through Automated Resume Screening with NLP and Machine Learning," in *Proc. 2024 IEEE International Conference on Intelligent Systems and Networking (INSPECT)*, 2024.
- [2] M. Strohmeier and F. Piazza, "Artificial Intelligence Techniques in Human Resource Management: A Conceptual Exploration," *IEEE Computational Intelligence Magazine*, vol. 18, no. 2, pp. 45–54, May 2023.
- [3] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Stanford, CA, USA: Stanford University, 2023.
- [4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proc. International Conference on Learning Representations (ICLR)*, 2013.
- [5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 4171–4186.
- [6] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O'Reilly Media, 2009.
- [7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.